Langara College
CPSC 2150
Instructor Gladys Monagan

Assignment #3: Complexity Analysis

Assignment due with Brightspace at 11:50pm on February 7

Read in the textbook by Drozdek chapter 2 (skip §2.10).
Read in the textbook by Goodrich et al. chapter 4 and §11.1.5

Exercise 1
Show that $8n^2 + 7n$ is $O(n^2)$

Exercise 2
Simplify $n\,O(7n^2 + 2)$
giving the properties (facts) that you are using.

Exercise 3
Simplify $O(55n + 2\log n + 9n)$
giving the properties (facts) that you are using.

Exercises 4 – 6 below give 3 program fragments. Find the complexity of each. Express the complexity in accurate O(f(n)) notation. Show your work.

To the right of `some code` you are given the number of operations in the code, either as exact operations or with big O notation.

Exercise 4

step 1 `some code` $O(n \log_2 n)$
step 2 `some code` $O(n^2)$
step 3 `some code` $O(n^2)$
step 4 `some code` $O(n)$

Exercise 5

```
step 1   [some code]  O(n log2n)
         for (i=0; i < n; i++)
         {
step 2a      [some code]  O(n)
step 2b      [some code]  O(1)
         }
step 3   [some code]  O(n)
```

Exercise 6

```
step 1   some code     n + 1
         for (i=1; i <= n; i++)
         {
step 2a      some code  i + 1
step 2b      some code  2
         }
step 3   some code   3n + 9
```

Exercise 7

Given

$$C_n = 2\,C_{n-1} + 2\,C_{n-2} + 2\,C_{n-3} \qquad C_1 = 1,\ C_2 = 1,\ C_3 = 1$$

determine C_4 and C_5

Exercise 8

Solve

$$C_n = C_{n-1} + 2n + 1 \qquad C_1 = 1$$

and afterwards check for C_3

Exercise 9

Given the function g

```
int g(int A[], int n)
{
   if (n == 1)
      return A[0];
   for (int i = 1; i < n; i++)
   {
      A[i-1] += A[i];
   }
   g(A, n-1);
   return A[0];
}
```

How many times is the statement `A[i-1] += A[i];` executed?
Give a recurrence relation with initial value.
Solve and give your answer in closed-form and then use the big O notation.
Show your work.

Exercise 10

Given the function

```
void h(int x, int y, int n) {
   if (n == 0) {
      std::cout << "n=0" << std::endl;
      return;
   }
   h(x - 1, y + 1, n - 1);
   std::cout << "n=" << n << "   " << x << " " << y << std::endl;
   h(y - 1, x + 1, n - 1);
}
```

Count the number of lines of output?
Give a recurrence relation with initial value.
Solve and give your answer in closed-form and then use the big O notation.
Show your work.

To submit as Assignment #3 a single pdf file (a single document)

Submit a pdf file with the Exercises 1 – 10 in order and put your name and student number on the first page of the document.
Please mark clearly which exercise is being solved.
Place the exercises in order.

You can write by hand and scan or photograph your solutions.
Solve the exercises in order. Please make sure that they are legible.
Or produce the document electronically e.g. with a tablet and submit a pdf.

And, of course, you could produce the answer using LaTex, MS Word or some other text processing program.
Produce a pdf and submit the pdf.

Marking Scheme out of 100

Exercise 1:	10 marks
Exercise 2:	5 marks
Exercise 3:	5 marks
Exercise 4:	8 marks
Exercise 5:	10 marks
Exercise 6:	10 marks
Exercise 7:	6 marks
Exercise 8:	16 marks
Exercise 9:	14 marks
Exercise 10:	16 marks